

**Amendments to the Claims:**

This listing of claims will replace all prior versions, and listings, of claims in the application:

**Listing of Claims:**

1. (Currently Amended) A method in a data processing system for collecting data for analyzing memory leaks, the method comprising:  
associating a plurality of indicators with a plurality of objects, wherein the plurality of indicators are set to a first state;  
setting an indicator for each live object in the plurality of objects from the first state to a second state; and  
initiating a garbage collection process to free unused objects; and  
responsive to a request for the data, collecting data from all objects in the plurality of objects having indicators set to the first state, wherein the data collected only contains data associated with the memory leaks and wherein the memory leaks occur when unused objects are not freed during the garbage collection process.
2. (Canceled)
3. (Original) The method of claim 1, wherein each of the plurality of indicators is a bit associated with a corresponding object in the plurality of objects.
4. (Original) The method of claim 3, wherein the first state is a logic zero and the second state is a logic one.
5. (Original) The method of claim 1, wherein the method is implemented in a Java virtual machine.
6. (Original) The method of claim 1, wherein the data is placed into a text file.
7. (Original) The method of claim 1, wherein the plurality of objects are located in a heap.
8. (Original) The method of claim 1, wherein the request is received from a Java application.

9. (Currently Amended) A method in a data processing system for collecting data used to detect memory leaks in a Java virtual machine, the method comprising:

associating an indicator with an object in a heap, wherein the indicator is set to a default state; responsive to a first request, setting indicators for all live objects in the heap from the default state to a live state; and

responsive to a second request, collecting information for all objects having indicators in the default state after a garbage collection process executes to free unused objects, wherein objects having indicators in the default state are objects with memory leaks.

10. (Original) The method of claim 9, wherein the indicator is associated with the object when the object is created.

11. (Original) The method of claim 9, wherein the first request and the second request are received from a Java application.

12. (Original) The method of claim 9, wherein the indicator set to the live state is a logic one and the indicator set to the default state is a logic zero.

13. (Currently Amended) A data processing system ~~in a data processing system~~ for collecting data for analyzing memory leaks, the data processing system comprising:

associating means for associating a plurality of indicators with a plurality of objects, wherein the plurality of indicators are set to a first state;

setting means for setting an indicator for each live object in the plurality of objects from the first state to a second state; and

initiating means for initiating a garbage collection process to free unused objects; and

collecting means, responsive to a request for the data, for collecting data from all objects in the plurality of objects having indicators set to the first state, wherein the data collected only contains data associated with the memory leaks and wherein the memory leaks occur when unused objects are not freed during the garbage collection process.

14. (Canceled)

15. (Original) The data processing system of claim 13, wherein each of the plurality of indicators is a bit associated with a corresponding object in the plurality of objects.
16. (Original) The data processing system of claim 15, wherein the first state is a logic zero and the second state is a logic one.
17. (Original) The data processing system of claim 13, wherein the method is implemented in a Java virtual machine.
18. (Currently Amended) A data processing system ~~in a data processing system~~ for collecting data used to detect memory leaks in a Java virtual machine, the data processing system comprising:
  - associating means for associating an indicator with an object in a heap, wherein the indicator is set to a default state;
  - setting means, responsive to a first request, for setting indicators for all live objects in the heap from the default state to a live state; and
  - collecting means, responsive to a second request, for collecting information for all objects having indicators in the default state after a garbage collection process executes to free unused objects, wherein objects having indicators in the default state are objects with memory leaks.
19. (Currently Amended) A computer program product in a computer readable medium for collecting data for analyzing memory leaks, the computer program product comprising:
  - first instructions for associating a plurality of indicators with a plurality of objects, wherein the plurality of indicators are set to a first state;
  - second instructions for setting an indicator for each live object in the plurality of objects from the first state to a second state; and
  - third instructions for initiating a garbage collection process to free unused objects; and
  - fourth third instructions, responsive to a request for the data, for collecting data from all objects in the plurality of objects having indicators set to the first state, wherein the data collected only contains data associated with the memory leaks and wherein the memory leaks occur when unused objects are not freed during the garbage collection process.
20. (Canceled)

21. (Original) The computer program product of claim 19, wherein each of the plurality of indicators is a bit associated with a corresponding object in the plurality of objects.

22. (Original) The computer program product of claim 21, wherein the first state is a logic zero and the second state is a logic one.

23. (Original) The computer program product of claim 19, wherein the method is implemented in a Java virtual machine.

24. (Currently Amended) A computer program product in a computer readable medium in a data processing system for collecting data used to detect memory leaks in a Java virtual machine, the computer program product comprising:

first instructions for associating an indicator with an object in a heap, wherein the indicator is set to a default state;

second instructions, responsive to a first request, for setting indicators for all live objects in the heap from the default state to a live state; and

third instructions, responsive to a second request, for collecting information for all objects having indicators in the default state after a garbage collection process executes to free unused objects, wherein objects having indicators in the default state are objects with memory leaks.

25. (Original) The computer program product of claim 24, wherein the indicator is associated with the object when the object is created.

26. (Original) The computer program product of claim 24, wherein the first request and the second request are received from a Java application.

27. (Currently Amended) A data processing system comprising:

a bus system;

a memory connected to the bus system, wherein the memory includes a set of instructions; and

a processing unit connected to the bus system, wherein the processing unit executes a set of instructions to associate a plurality of indicators with a plurality of objects, wherein the plurality of indicators are set to a first state; set an indicator for each live object in the plurality of objects from the first state to a second state; initiate a garbage collection process to free unused objects; and collect data

from all objects in the plurality of objects having indicators set to the first state, in response to a request for the data, wherein the data collected only contains data associated with the memory leaks and wherein the memory leaks occur when unused objects are not freed during the garbage collection process.

28. (Currently Amended) A data processing system comprising:

- a bus system;
- a memory connected to the bus system, wherein the memory includes a set of instructions; and
- a processing unit connected to the bus system, wherein the processing unit executes a set of instructions to associate an indicator with an object a heap, wherein the indicator is set to a default state; set indicators for all live objects in the heap from the default state to a live state, in response to a first request; and collect information for all objects having indicators in the default state after a garbage collection process executes to free unused objects, wherein objects having indicators in the default state are objects with memory leaks, in response to a second request.